

# HW2

---

T1

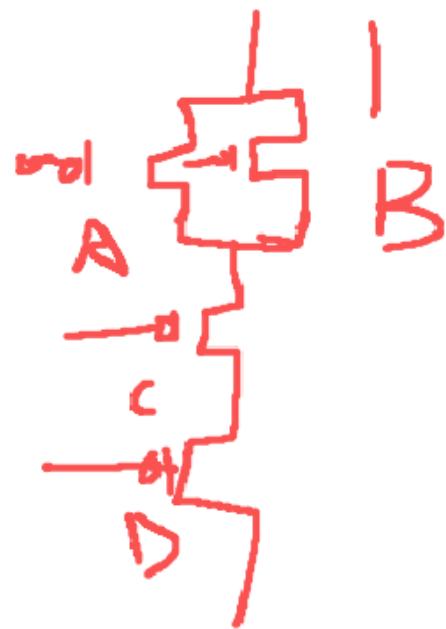
x  
1  
0  
x  
0

T2

1

A	B	C	Out
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

2



T3

1:NOR 门本身足够表示 NOT、OR 和 AND 操作

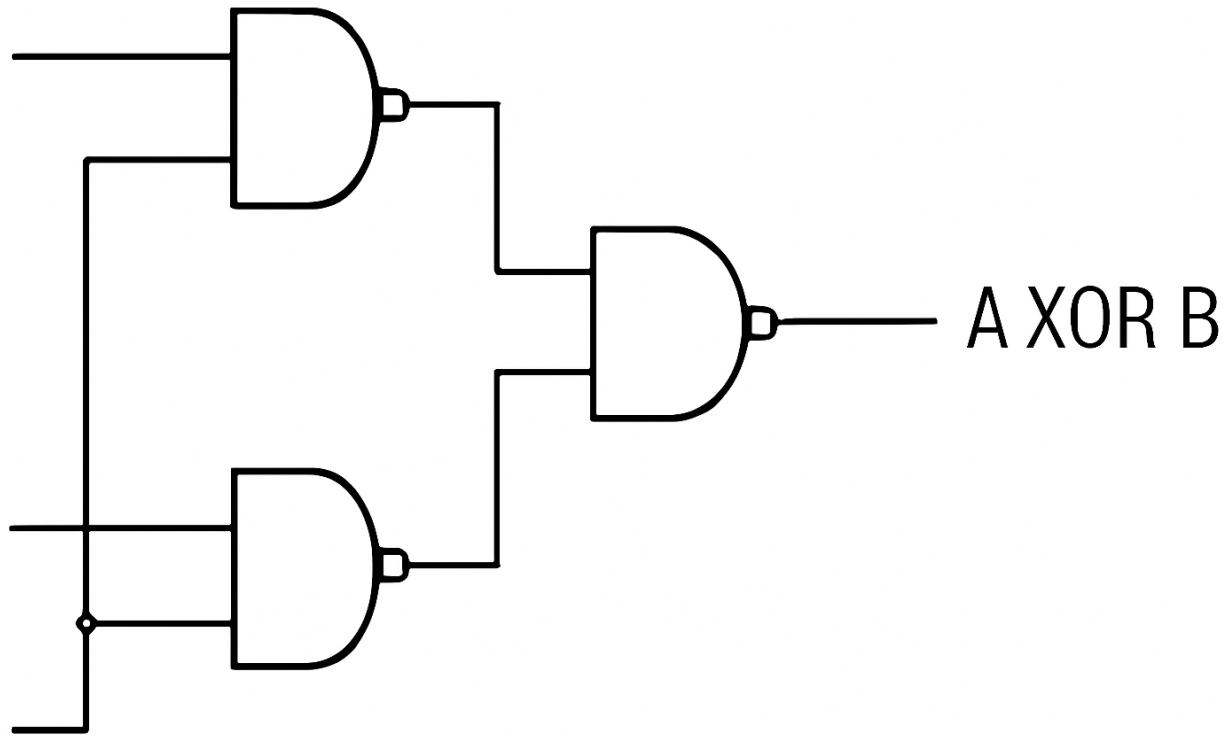
2:异或门无法表示 NOT 门，而 NOT 门是构成其他逻辑运算的基本部分，因此异或门 在逻辑上不是完备的。

T4

1. $2^{-126}$

2. $2^{-126} \sim 1.17549435 \times 10^{-38}$

T5



T6

1

0 0 1

2

寻址空间：指微处理器能够寻址的内存单元的总数量。寻址能力：指微处理器一次可以访问（读取或写入）的数据位数 4位

3

扩展数据总线：将数据线的数量从 4 条 ( $D[3:0]$ ) 增加到  $k$  条 ( $D[k-1:0]$ )。

扩展每个存储单元：内存阵列中的每一个存储单元的宽度（容量）需要从 4 位相应地扩展到  $k$  位，以便每个单元能存储  $k$  位数据。

补充说明（寻址空间的扩展）

T7

-32768 到 32767。

T8

冯·诺伊曼模型的主要组成部分包括控制器、运算器、存储器、输入设备和输出设备，其核心思想是采用二进制表示信息，并将程序与数据存储在同一存储器中，通过存储程序控制实现指令的顺序执行（除非遇到跳转指

令）。然而，量子计算机不能直接应用该模型，因为量子计算机依赖量子比特的叠加和纠缠特性，允许并行处理多种状态，而冯·诺依曼架构基于经典二进制比特（0或1）的串行处理；此外，量子不可克隆定理限制了程序的直接存储和复制，且量子信息易受环境干扰（退相干），导致传统内存访问方式（如共享总线）无法适配量子系统的独特需求。

## T9

1:011111

2: 12

3:12

4 :12

## T10

当前状态	投1元 (I1)	投5元 (I5)
S0	S1	S5
S1	S2	S6
S2	S3	S7
S3	S4	S0 (发货)
S4	S5	S0 (发货)
S5	S6	S0 (发货)
S6	S7	S0 (发货)
S7	S0 (发货)	S0 (发货)

## T11

将四只老鼠视为四个二进制位，分别代表权重1、2、4、8（即老鼠A、B、C、D），并将八瓶酒编号为1至8。每瓶酒的编号用4位二进制表示（如酒瓶1为0001，酒瓶2为0010等），根据二进制位为1的位置，让对应老鼠喝这瓶酒（例如，酒瓶1只有老鼠A喝，酒瓶3有老鼠A和B喝）。测试后，将死亡老鼠的权重相加，所得数字即为毒酒编号（如只有老鼠A死亡，和为1，对应酒瓶1有毒），从而唯一确定毒酒。

在这个策略中，四只老鼠最多能检测16瓶酒是否有毒。因为四只老鼠的生死状态有 $2^4=16$ 种可能组合（从无死亡到全部死亡），每种组合可以唯一对应一瓶酒。通过二进制编码，策略能充分利用老鼠的死亡模式来区分酒瓶，但如果酒瓶数超过16，则无法保证唯一区分。

在这个策略中，预计死亡老鼠数量的期望值为1.625只。计算基于每瓶酒有毒概率相等（1/8），死亡数等于毒酒编号二进制中“1”的个数：酒瓶1、2、4、8各死1只（概率4/8），酒瓶3、5、6各死2只（概率3/8），酒瓶7死3只（概率1/8），期望值 =  $(1 \times 4/8 + 2 \times 3/8 + 3 \times 1/8) = 13/8 = 1.625$ 。可以修改策略减少预期死亡数量，例

如只使用三只老鼠 (A、B、C) 检测八瓶酒，让老鼠D不参与，这样死亡模式000对应酒瓶8有毒，其他模式对应酒瓶1-7，期望死亡数降至 $(0 \times 1/8 + 1 \times 3/8 + 2 \times 3/8 + 3 \times 1/8) = 12/8 = 1.5$ 只，但这会牺牲检测能力，因为三只老鼠最多只能检测8瓶酒。